

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1 1. (Currently Amended) A method of optimizing a query in a multi-tenant
2 database, said multi-tenant database having one or more data tables, each table having one or
3 more logical columns defining data categories and one or more logical rows associated with one
4 or more tenants, wherein a plurality of tenants have data stored in the data tables, the method
5 comprising:
6 determining database indices for one or more of the data tables;
7 generating tenant-level statistics for one or more of said plurality of tenants for
8 one or more of the data tables;
9 receiving a SQL query; and
10 optimizing the SQL query based on the database indices; and
11 optimizing the SQL query based on the tenant-level statistics, thereby enabling SQL query
12 optimization according to greater semantic knowledge of use of the data tables.

1 2. (Original) The method of claim 1, wherein each tenant includes one or more
2 associated users, the method further including:
3 generating user-level statistics for one or more of the users of one or more of the
4 tenants for one or more of the data tables; and
5 optimizing the SQL query based on the user-level statistics.

1 3. (Original) The method of claim 2, wherein the user-level statistics are stored
2 to a user metadata table.

1 4. (Original) The method of claim 2, wherein generating user-level statistics
2 includes determining a total number of distinct rows for each of said plurality of users.

1 5. (Original) The method of claim 4, wherein the total number is an approximate
2 number based on one or more of a) a number of rows viewable by the user and users below the

3 user in a role hierarchy, b) a number of rows that are shared by a group to which the user belongs
4 and c) a number of rows that are manually shared to the user by another user or group of users.

1 6. (Original) The method of claim 2, wherein generating user-level statistics for
2 a user is performed according to one of a) on a scheduled basis, b) after a predetermined number
3 of queries by the user, and c) each time an unconstrained query is run by the user.

1 7. (Original) The method of claim 1, wherein generating tenant-level statistics is
2 performed on a periodic basis.

1 8. (Original) The method of claim 1, wherein generating includes determining a
2 total number of distinct rows accessible for each of said plurality of tenants.

1 9. (Original) The method of claim 8, wherein the tenant-level statistics are stored
2 to a tenant metadata table.

1 10. (Original) The method of claim 1, wherein at least one column of one of said
2 tables includes data associated with two or more tenants.

1 11. (Currently Amended) A multi-tenant database system, comprising:
2 a database having one or more data tables, each table having one or more columns
3 defining data categories and one or more rows associated with one or more tenants, wherein a
4 plurality of tenants have data stored in the data tables;

5 a database indices determining module configured to generate database indices for
6 one or more of the data tables of the multi-tenant database;

7 a tenant-level statistics generating module configured to generate tenant-level
8 statistics for one or more tenants for one or more of the data tables; ~~and~~

9 a database indices optimization module, configured to optimize a database query
10 on the database indices; and

11 a tenant-level query optimization module, configured to optimize a database
12 query based on the tenant-level statistics.

1 12. (Original) The multi-tenant database system of claim 11, wherein each tenant

2 includes one or more associated users, wherein the statistics generating module is further
3 configured to generate user-level statistics for each user, and wherein the query optimization
4 module is further configured to optimize the database query based on the user-level statistics.

1 13. (Original) The system of claim 12, further including a memory module,
2 wherein the statistics generating module stores the user-level statistics to a metadata table in the
3 memory module.

1 14. (Original) The system of claim 12, wherein the statistics generating module generates
2 user-level statistics by determining a total number of distinct rows for each of said plurality of users.

1 15. (Original) The system of claim 14, wherein the total number is an
2 approximate number based on one or more of a) a number of rows viewable by the user and
3 users below the user in a role hierarchy, b) a number of rows that are shared by a group to which
4 the user belongs and c) a number of rows that are manually shared to the user by another user or
5 group of users.

1 16. (Original) The system of claim 12, wherein the statistics generating module
2 generates user-level statistics for a user according to one of a) on a scheduled basis, b) after a
3 predetermined number of queries by the user, and c) each time an unconstrained query is run by
4 the user.

1 17. (Original) The system of claim 11, wherein the statistics generating module
2 generates tenant-level statistics on a periodic basis.

1 18. (Original) The system of claim 11, wherein the statistics generating module generates
2 tenant-level statistics by determining a total number of distinct rows viewable for each of said plurality of
3 tenants.

1 19. (Original) The system of claim 18, further including a memory module,
2 wherein the statistics generating module stores the tenant-level statistics to a tenant metadata table
3 in the memory module.

1 20. (Original) The system of claim 11, wherein at least one column of one of said
2 tables includes data associated with two or more tenants.

1 21. (Currently Amended) A method of optimizing a query in a multi-tenant
2 database, said database having one or more data tables, each table having one or more logical
3 columns defining data categories and one or more logical rows associated with one or more
4 tenants, wherein a plurality of tenants have data stored in the data tables, and wherein each tenant
5 includes one or more users, the method comprising:

6 processing the data tables so as to determine tenant-level statistics for each of said
7 plurality of tenants;

8 processing the data tables so as to determine user-level statistics for each of said
9 plurality of users;

10 receiving a SQL query; and

11 optimizing the SQL query based on one or both of the tenant-level statistics and the
12 user-level statistics.

1 22. (Original) The method of claim 21, further including:

2 storing the user-level statistics to a user-level metadata table in a memory module;
3 and

4 storing the tenant-level statistics to a tenant-level metadata table in the memory
5 module.

1 23. (Original) The method of claim 21, wherein determining user-level statistics
2 includes determining a total number of distinct rows for each of said plurality of users, and
3 wherein determining tenant-level statistics includes determining a total number of distinct rows
4 for each of said plurality of tenants.

1 24. (Original) The method of claim 21, wherein processing the data tables to
2 determine tenant-level statistics is performed on a periodic basis.

1 25. (Original) The method of claim 21, wherein processing the data tables to

2 determine user-level statistics for a user is performed according to one of a) on a scheduled basis,
3 b) after a predetermined number of queries by the user, and c) each time an unconstrained query
4 is run by the user.

1 26. (Previously Presented) The method of claim 1, wherein the generating
2 tenant-level statistics comprises generating tenant-level statistics for each of said plurality of
3 tenants for each of the data tables.

1 27. (Previously Presented) The method of claim 2, wherein the generating user-
2 level statistics comprises generating user-level statistics for each of the users of each of the tenants
3 for each of the data tables.

1 28. (Previously Presented) The multi-tenant database system of claim 11, wherein
2 the statistics generating module is configured to generate tenant-level statistics for each tenant for
3 each of the data table.

1 29. (New) The method of claim 1, further comprising:
2 determining a particularity of a query being executed; and
3 conducting at least one of optimizing the SQL query and generating statistics based on the
4 particularity of the query being executed.

1 30. (New) The method of claim 1, further comprising:
2 determining application information corresponding an application for which the
3 multi-tenant database is used; and
4 optimizing the SQL query based on the application information.

1 31. (New) The method of claim 30, wherein the application information includes
2 data sharing among at least two users of at least one tenant.

1 32. (New) The method of claim 31, wherein the optimizing the SQL query based
2 on the application information includes determining whether to access a database object from a
3 user side or a tenant side.

1 33. (New) The method of claim 1, wherein the optimizing the SQL query based on
2 the tenant-level statistics modifies the optimization of the SQL query based on the database
3 indices, thereby enabling the tenant-level statistics to tune query optimization otherwise provided
4 by the optimizing the SQL query based on the table-level statistics.

1 34. (New) The multi-tenant database system of claim 11, wherein the tenant-level
2 query optimization module modifies the optimization of the SQL query by the database indices
3 optimization module, thereby enabling the tenant-level statistics to tune query optimization
4 otherwise provided by the table-level query optimization module.